AD-A155 527

(2)

A RAND NOTE

FUTURE MILITARY APPLICATIONS
FOR KNOWLEDGE ENGINEERING

Steven C. Bankes

February 1985

N-2102-AF

Prepared for

The United States Air Force

Rand
1700 MAIN STREET
P.O. BOX 2138
SANTA MONICA, CA 90406-2138

The research reported here was sponsored by the Directorate of Operational Requirements, Deputy Chief of Staff/Research, Development, and Acquisition, Hq USAF, under Contract F49620-82-C-0018. The United States Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon.

Published by The Rand Corporation

A RAND NOTE

FUTURE MILITARY APPLICATIONS FOR KNOWLEDGE ENGINEERING

Steven C. Bankes

February 1985

N-2102-AF

Prepared for

The United States Air Force

Accession For	
NTIS GRA&I	X
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A/1	

Rand
1700 MAIN STREET
P.O. BOX 2138
SANTA MONICA, CA 90406-2138

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER N-2102-AF	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Future Military Applicatons for Knowledge Engineering		5. TYPE OF REPORT & PERIOD COVERED Interim
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Steven C. Bankes		8. CONTRACT OR GRANT NUMBER(s) F49620-82-C-0018
9. PERFORMING ORGANIZATION NAME AND ADDRESS The Rand Corporation 1700 Main Street Santa Monica, CA 90406		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Requirements, Programs & Studies Group (AF/RDQM) Ofc, DCS/R&D and Acquisition Hq USAF, Washington, DC 20330		12. REPORT DATE February 1985
		13. NUMBER OF PAGES 51
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release: Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

No Restrictions

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Artificial Intelligence
Decision Making
Computer Applications

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

See reverse side

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

This Note surveys the technology of knowledge-based systems, or knowledge engineering, to identify potential military applications. The number of potential applications is quite large, and includes: decision support systems, knowledge-based simulations, training aids, maintenance advisors, robotic applications. Because the technology is new and untested, there are a number of risk factors: It is unclear how costs will increase with the size of any given project; debugging and verification may be more difficult than in traditional software engineering; support resources (hardware, software, and skilled personnel) will have to be developed before this technology can be widely applied. To facilitate the parallel development of fundamental research and practical development, the author recommends selecting key applications for actual system development and, to avoid costly failures, suggests that an evolutionary strategy be employed. Knowledge-based systems should be designed to augment human capabilities, not replace personnel.

PREFACE

This Note was produced under the Project AIR FORCE study effort "Air Force Computer Systems Applications." It examines how anticipated improvements in computer technology will affect the Air Force mission over the next 25 years. The design of software for complex symbolic tasks is one area where great improvement is expected. Techniques pioneered by the artificial intelligence community hold the promise of performing tasks heretofore requiring human judgment. Knowledge engineering, the application of these techniques to real problems, while still in its infancy, has attracted much attention owing to the greatness of its potential. The following pages review the status of knowledge engineering and survey its applicability to a variety of military domains.

The author would like to thank John Clark, Jean Gebman, and Willis Ware, who made numerous helpful comments on an early draft, and also David Leinweber, who contributed several ideas.

SUMMARY

This Note surveys the technology of knowledge based systems, or knowledge engineering, to identify potential military applications. The number of potential applications is quite large. Possible capabilities that could be gained from advances in knowledge engineering include:

- Providing a means for collecting, storing, and distributing the accumulated knowledge of experts.
- Providing a technique for creating computer software that can perform functions too complex for algorithmic treatment.
- Allowing greater access to computerized resources by untrained personnel.
- Allowing the creation and management of systems whose complexity might otherwise be prohibitive.
- Allowing the construction of mission equipment with far more capabilities than any equipment now has.

Some of the more specific applications for knowledge engineering technology might include:

- Decision support systems
- Knowledge based simulations
- Training aids
- Maintenance advisors
- Eventual robotic applications

It is important to note that while it is extremely promising, knowledge engineering is a novel technology. Many substantial problems remain to be solved before it can be successfully applied. Important risk factors associated with this novel technology include:

- It is relatively untested. There are likely to be unexpected problems in its application of any given domain.
- It is an immature technology. Further fundamental research is needed, the progress of which is difficult to predict.
- There are unresolved scaling problems--e.g., how will costs increase with the size of any given project. These issues are both technological (e.g., how does increasing the size of the domain increase the complexity of the software required?) and managerial (e.g., can the tasks of knowledge engineering be standardized to permit large manpower efforts?).
- Debugging and verification issues may be more difficult than in traditional software engineering. At the very least, new approaches to these issues will have to be developed.
- Support resources will need to be developed before this technology can be widely applied. These include hardware and software, but the critical shortage is likely to be skilled personnel--programmers, system designers, and managers.

To facilitate the parallel development of fundamental research and practical development, it would be desirable to select key applications for actual system development. Limited domain computerized assistants and knowledge-based simulations are two areas that might be considered, since demonstrations of principle already exist for them.

As with any novel technology, there is potential for initial disappointment. To avoid costly failures, we suggest that an evolutionary strategy be employed. Applications should be carefully chosen both for technical criteria and for the potential to enhance capabilities without becoming indispensable. We further suggest that all initial projects be well defined and bounded. We believe that knowledge-based systems should be designed to augment human capabilities, not replace personnel. Knowledge engineering holds perhaps the greatest promise for creating computer software that is flexible enough to provide people with enhanced capabilities and still preserve all their options.

CONTENTS

I. INTRODUCTION

The academic field of Artificial Intelligence (AI) studies computational mechanisms which allow machines to perform tasks which previously were performed only by humans. The long-term goal of constructing an "artificial intelligence" is still primarily of philosophical rather than practical interest. However, artificial intelligence is also a part of advanced computer science--making computer systems behave in ways which were heretofore impossible. Consequently, techniques pioneered by the artificial intelligence community have periodically found widespread practical application. Time-shared interactive systems and man-machine interfaces using windows and mice are two examples of such spinoffs.

The current popularity of artificial intelligence dates from the success of a class of consulting programs, the so-called expert systems which imitate the judgments of human experts in given domains of expertise. Examples include systems that can diagnose diseases, perform complex symbolic calculations, interpret geophysical and sonar data, and make expert legal judgments. These systems make use of information obtained through extensive interactions with experts to imitate the experts' reasoning process when it confronts a problem. After repeated system testing and expert debriefing, the program's behavior can be made to closely resemble the expert's, yielding impressive results [3,4,7,13,17,25].

In the vernacular of the field, these systems are said to be "knowledge-based," and the techniques for constructing them are called "knowledge engineering"[14]. Knowledge engineering can immediately be seen to be a superior term to applied artificial intelligence. It is less inflammatory or misleading as well as more descriptive. As this technology becomes established, knowledge engineering may become the dominant term and artificial intelligence will again be used to refer to basic research into very difficult problems.

There are many potential applications for knowledge engineering beyond that of expert systems. Artificial intelligence systems deal symbolically with a wide range of entities and the relationships between them. Thus any task with a large symbolic component should be considered as a candidate for a knowledge based approach. One example of such a potential application would be a system capable of understanding natural languages.

The set of techniques that constitute knowledge engineering can be viewed as programming devices for dealing with extremely complex problem spaces. The domains best suited to knowledge engineering (1) exist in complex physical or social contexts, and (2) do not yield to algorithmic solutions, and require instead very situation-dependent means such as rules of thumb, searches for solutions among plausible possibilities, and reasoning from partial information. While knowledge engineering techniques differ from more familiar programming methods, the development of practical knowledge based systems will share many characteristics with traditional software engineering.

Knowledge engineering can thus be viewed as a branch of software engineering. It is a branch concerned with symbolic information processing, especially those tasks lacking an algorithmic solution that would make them amenable to more traditional approaches. These tasks typically are non-numeric and lack a regular structure. Knowledge engineering as it exists today can solve only a very small percentage of such problems, but even where it can not, it represents the best presently available approach to their solution. If these problems are to be solved, it seems likely that knowledge engineering will be the approach.

The remainder of this document will consider opportunities for the utilization of this technology. This material should assist the reader in assessing whether a particular problem is a likely candidate for a knowledge based approach. Section II describes the present state of technology in more detail. Section III describes several general categories of applications. Section IV examines special considerations distinguishing military applications, and Section V suggests some specific applications of the technology to problems of importance to the

Air Force. Section VI discusses some difficulties inherent in applying this technology to large problems. Section VII considers applications to command, control, and communications. Section VIII discusses applications to robotics, machine vision, and natural language understanding. Section IX considers the prospects for computers which are specially designed to support knoweldge based applications, and Section X reviews the major conclusions of this Note.

II. THE PRESENT STATE OF THE TECHNOLOGY

Knowledge engineering is a young and rapidly changing field which is at present as much an art as a science. A tutorial exposition is far beyond the scope of this Note, but excellent technical references as well as general introductions can be found elsewhere [3,13,17,18,20,36,38]. These are reference works or texts that provide a fairly comprehensive technical treatment. For an introductory article the reader may wish to consult Waltz, "Artificial Intelligence," *Scientific American*, October 1982 [36]. A source book on artificial intelligence research is Barr and Feigenbaum, *Handbook of Artificial Intelligence* [3]. An in-depth treatment of expert systems can be found in Hayes-Roth, Waterman, and Lenat, *Building Expert Systems* [20]. Additional technical references may be found in the References.

The raw material of knowledge engineering is knowledge. The knowledge may be facts about the domain or situation-specific problem solving methods. Knowledge engineering involves obtaining this knowledge (which in many cases means extracting it from a human expert), structuring it in a way which allows for its representation, and organizing the process for the utilization of the knowledge. The technique for doing each of these three tasks may vary from domain to domain.

KNOWLEDGE ACQUISITION

Knowledge acquisition is the process of obtaining necessary knowledge for the successful completion of a knowledge based system. Some knowledge may exist in easily available forms such as reference texts or electronic data bases. A much more difficult source that is often critical is information in the minds of certain human experts. Such experts may provide rules of thumb, subtle discriminations of judgment, or other forms of compiled experience. Often these do not exist as preconceived rules but must be devised by working through examples or other interactive processes. Where the needed knowledge does not exist, it must be created, which involves a research effort in its own right.

The process of drawing knowledge from a human expert is critical to the construction of an expert system. This process is typically very time consuming and involves many iterations. The expert must also help debug the sets of rules he provides when they do not produce the same judgment that he would make. Tools to assist this endeavor may eventually be created. However, the process of knowledge acquisition will remain labor intensive, especially in the application of the technology to new domains.

KNOWLEDGE REPRESENTATION

The task of knowledge representation is to devise a regular form for the knowledge which has been obtained so that it may be stored and manipulated on the computer. The problem of the representation of knowledge is one of the central issues in artificial intelligence, and the current set of knowledge representation forms constitutes the frontier of knowledge engineering. Knowledge which cannot be adequately represented cannot be utilized. Each system will tend to have a slightly different form of knowledge representation, but in general there are a few general forms which are used.

Knowledge which is easily regularized is data, which can be dealt with adequately with conventional programming technology. Knowledge which cannot be regularized is not utilizable with present technology. The available techniques of knowledge representation are all methods of placing irregular information in an externally regular form, thus allowing the information to be dealt with via a uniform mechanism. Examples of knowledge representation methods are IF-THEN rules (sometimes called productions), semantic networks, frames, and scripts [3,17,20,36]. New techniques for knowledge representation will undoubtedly arise as the field evolves. Because each method makes certain kinds of knowledge easier to represent than others, the choice of knowledge representation (which must be made early in expert system development) is important [39].

KNOWLEDGE UTILIZATION

The knowledge based system, whatever its purpose, must select from the various pieces of knowledge it possesses those few which, when combined, yield the desired result. For a given problem statement, the pieces of information which are potentially relevant may greatly exceed those which ultimately will be used. Thus the system may need to make several attempts at problem solution before the correct combination is discovered. Consequently, the process used for problem solution may involve searching through a space of solution possibilities. For example, in the case of a rule based system, a solution to a given problem may involve a specific chain of rule applications. Thus the system may try starting at the problem statement and working forward toward the solution by trying various possibly relevant rules. Or it may work backwards from the goal (such systems are called backwards chaining). In either case, if the system selects rules at random, it will not finish in reasonable time except for very simple problems. The knowledge based system must thus be able to strongly focus the search--trying very likely possibilities first, with false starts held to a minimum. The knowledge concerning how to look for a solution may be called meta-knowledge. Meta-knowledge may exist only as embodied in the program itself, or it like other knowledge may be represented in some explicit form.

EXPERT SYSTEMS

Expert systems are at present the principal application of knowledge engineering technology. Such systems reason from knowledge in a limited domain in a way that mimics the judgment of experts in that domain, hence the name, expert systems. Some expert systems have been demonstrated in tests to perform competitively with human experts [34].

Such systems are potentially useful for a number of reasons. Human experts are often in short supply. If the problem solving expertise of a single expert or team of experts can be captured in a computer program, then this knowledge can be replicated and placed in any location in which it is needed. A human with less than expert knowledge assisted by such a system can make judgments and otherwise perform at a

level comparable to that of a much more skilled individual. Such systems, if provided with means to implement their decisions, can be placed in inhospitable environments and potentially may handle data at speeds impossible for a man. Expert systems allow expertise to be retained, safe against the retirement or death of the source expert. They are the primary current example of a style of computer programming which we have chosen to refer to as knowledge engineering.

AREAS OF CURRENT DIFFICULTY

Although the recent successes of knowledge engineering have been encouraging, there remain a number of situations which have yet to be adequately dealt with. The following is a sampling of such problem areas. The list attempts to capture some of the major fronts where further research will be required if the technology is to be applied to a wide range of problem areas. It is meant to be suggestive, not exhaustive.

One great need is for methods of representing and reasoning about time dependent information. The problem has been approached in basic research, but there has yet to be a major application successfully incorporating time dependent knowledge. This representational issue, coupled with difficulty in managing highly dynamic data, limits the application of knowledge based technology to real world domains. There appears to be no real barrier to solving this difficulty; however, its solution will require substantial fundamental research, and some good ideas.

Further work may also be appropriate in the representation of uncertainty. Many different approaches have been utilized in existing systems [3]. However, many are rather ad hoc, will not easily generalize to new domains, and introduce fragile tuning requirements which hamper system maintenance. There is also a significant knowledge acquisition problem in regard to plausibility estimates. Additionally, the cross product of uncertainty with other features, such as time representation, is generally uninvestigated.

Planning is a task area which has received attention [3], but is still in need of further innovation. Methods for decomposing complex planning tasks into nearly optimal pieces are needed, as are further improvements in detecting and correcting subgoal interaction.

Knowledge acquisition can be identified as a potential bottleneck in the knowledge engineering process. Consequently, tools to assist in this task, as well as methods for automating it, would prove very beneficial. Such tools could range from managerial techniques for this task to knowledge based knowledge acquisition aids.

The greatest gains to be had from this technology may potentially come not from stand-alone knowledge based systems, but rather from using such systems to enhance the capabilities of humans. In order for man-computer teaming to be effective, the interface must be facile. The implications of this requirement are not well understood.

Additionally, there are multiple nontechnical barriers to the full exploitation of this technology. Most critically, there is a tremendous shortage of skilled individuals to perform the needed work. There is also a pronounced need for programmers skilled in the technology. The task of managing a large knowledge engineering development project may differ in important ways from more general software development efforts. These new principles will emerge only through the experiences of the first large projects. There will undoubtedly be a shortage of skilled technical managers with both the necessary technical expertise and professional maturity. Over the last decade, the difficulties associated with software engineering have been ameliorated through the design of tools, both software and managerial, to assist the process. The same development may be needed for knowledge engineering with special tools created to match its unique characteristics.

III. GENERIC APPLICATIONS

CHARACTERISTICS OF POTENTIAL APPLICATION AREAS

While there have been successful applications of knowledge engineering, the lack of experience in applying this technology is such that each such success must be regarded as something of a special case. Consequently, care must be taken in chosing the application to be addressed in near-term knowledge based projects. The greatest restriction on the application of expert systems is that the domain must be constrained. If the domain of interest is limited and closed, then the body of knowledge required for competency in this domain may be manageable. At the other extreme, any problem requiring "common sense" is at present impossible to handle adequately. Common sense turns out to constitute an enormous body of information. Somewhat ironically, designing systems to make judgments which are easy for a child has turned out to be much more difficult than systems for medical diagnosis, electrical circuit analysis, or mission planning.

Another requirement for the successful implementation of an expert system is that there must be an accessible source for the needed knowledge. In the case of common sense, the knowledge used to make such judgments is not held consciously. People are unable to list the knowledge they have used in solving an everyday task, and consistently underestimate the complexity of the knowledge required. Thus in addition to taking care that the amount of knowledge required be manageable, it is also necessary that there exist some source for the needed information. Typically this source is a human expert. This expert must be capable in the domain of interest. Additionally, his capability must be expressible in some concrete form. The expert must be able, with assistance, to articulate his special knowledge and to explain how it is applied to tasks of interest.

Thus, the first two general rules for picking a domain for the application of knowledge engineering may be expressed as follows. There must be an adequate source of knowledge accessible, and the knowledge required must be well bounded. Neither of these requirements can be expected to be significantly relaxed in the near future.

A third requirement for a good application area is that the knowledge required must be of a form for which a known representation method is applicable. This is necessary if the knowledge is to be placed in a regular form on which known inference methods may be employed. The requirement is a limiting one at the present time, but may become much less so as the expertise of the knowledge engineer increases during the next decade. There is tremendous variation in the ease with which knowledge can be represented. At the present time knowledge with explicit IF-THEN consequences is most easily used in an expert system.[1] As the field progresses, it can be expected that the range of knowledge which experience has taught us how to deal with will greatly increase.

In addition to the preceding necessary conditions, there are other general features of a good problem. To be a good economic choice, the problem must be intractable by more traditional methods and possess large economic leverage. The best candidates are problems where even a small improvement over current approaches would be important and where it is not necessary that the program function perfectly 100 percent of the time. This last indicator may become less important as the technology matures. Situations where the turnover of skilled personnel is a major problem may benefit from an expert system. Similarly, problem areas where skilled personnel are rare may be candidates. Other possible applications are those which have reasonably straightforward knowledge requirements but large combinatoric complexity. The general task which should be considered is one where people do well, algorithms do poorly, and where there exists a source of heuristic knowledge about the solution of the task.

---

[1] An example of an IF-THEN rule is:

```
If there is an airfield
    whose bomber-count is greater than 20
    and whose helicopter count is greater than 30
Then select-target the airfield.
```

(taken from Callero et al., *Toward an Expert Aid for Tactical Air Targeting* [7]).

At the present time, the set of potential applications for expert systems is constrained. The extent to which and the speed with which this technology can be expected to mature is not easy to estimate. However, until the maturation occurs, there will be a need to carefully select applications to ensure a reasonable chance of success.

A SAMPLER OF APPLICATIONS

A later section will consider specific military applications of this technology; here a more general overview of potential uses is provided to those considering the utilization of knowledge engineering. Each of the following application ideas could be used for a variety of problems. Consequently, each of these ideas can be considered relative to a particular problem area, yielding a spectrum of starting points for creative thinking.

Applications may be divided into classes of manned and unmanned systems, of which the former is the more substantial portion. Under manned applications we will mention automated decision aids, intelligent assistants, training aids, and knowledge based simulations. Unmanned systems will include both robotic applications and electronic agents.

Automated Decision Aids

The typical expert system is one which serves as a consultant. Such systems allow decisionmakers to have access to a collection of expert knowledge. The consultation system, upon being informed of the specifics of a situation, may ask questions and eventually give its opinion or recommendation. This advice could be adopted by the decisionmaker directly, or could provide input to his decision process. Ideally such a system would have a sophisticated explanation facility so that it could explain its reasoning in detail. Then unexpected advice could result in an interactive discussion between the human and his electronic advisor.

Similar systems could also be useful in assisting lower level personnel. As technology creates machines and environments which are increasingly complex, the problem of training humans to operate or repair that equipment grows increasingly severe. Thus, there is a need

for systems to advise the users and maintainers of this equipment. Such systems should accommodate user error, conform their behavior to the user's needs, and ideally would interact with the user in natural language. Such user friendly systems can allow less skillful personnel to adequately deal with the increasingly complex environments we are creating.

Intelligent Assistants

A step beyond consultation systems are systems which not only give advice, but which actively perform lower level functions, leaving the human free to perform those functions which he is best at. These intelligent assistants might have access not only to a knowledge store but also to active electronic data bases and communication facilities. Such systems could filter input, monitor volatile data, alert the user to events of interest, and fill out the details in plans and orders which the user need only sketch out.

Training Aids

Among the early applications of knowledge engineering, one which may have the greatest impact is the use of systems such as those above for training purposes. Even while prototype systems which are too slow or uneven in their performance to be used by active personnel are being refined, they may prove useful for training. Such trainers can place students in simulations of real case problems, and avail them of expert advice for coaching. As the world becomes increasingly complex, the burden on training will grow. Knowledge based training aids may help meet this very critical need.

Intelligent Simulation

Another facility of possible use to a decisionmaker would be an intelligent simulation facility. As opposed to numerical simulations, this model would represent the logical structure of organizations, physical situations, or human individuals. Such a system can serve as a "what-if?" machine, which would give answers to questions such as "what if I were to do this?" It would allow a decisionmaker to explore the simulated ramifications of various ideas before implementing them.

Simulation facilities can also be useful for training. Knowledge based techniques can model situations ranging from complex machines to the behavior of a potential adversary.

Robotics

Knowledge based systems allow for the creation of computer software to perform functions whose complexity was heretofore prohibitive. Among the applications this opens up are those using an intelligent machine to perform a task previously requiring a human. Tasks can thus be performed which otherwise would be too dangerous or inaccessible. Progress in robotic applications requires solving many problems not related to knowledge engineering. However, this technology has a contribution to make to the control of such devices.

Electronic Agents

Systems to monitor and effect external realities may be especially attractive in an electronic context. For example, systems to manage networks of computers, databases, or communications may be useful. Long before robotics technology is mature tasks in electronic domains may be performed by knowledge based agents. Among the functions such a system might perform are: monitoring data bases and alerting users of changes of interest or importance, and filtering or combining sources of information. In some situations actions may be taken without consulting an operator.

Knowledge Archival

Knowledge lost when experienced personnel change positions or retire is a great resource drain. An advanced idea is the use of expert systems technology to capture such expertise. This might eventually allow for the archiving of knowledge over the years. Such knowledge could be actively usable by consultation systems, as opposed to the passive knowledge collected in books and writings. It is questionable whether the technology for knowledge acquisition will ever mature sufficiently to allow for large-scale use of this idea. However, the debriefing of certain extremely useful experts so that their experience is not totally lost may be feasible. Even when such knowledge is not

used in an expert system, the rigor of placing the knowledge in an explicit form may be valuable.

IV. SPECIAL CONSIDERATIONS IN MILITARY AI APPLICATIONS

Deception is central to warfare. Modern warfare, with its proliferation of electronic systems, sensors, and countermeasures, is a particularly rich domain in which to practice this ancient military art. The complexity of the deceptive maneuvers possible in today's dense electronic environment is almost overwhelming, and, unfortunately, largely speculative, since many of the systems can never be tested, except in actual use.

All of the AI systems that have been built have dealt with honest environments. No one has deliberately falsified the information provided to any of them. Even without deliberately false information, coping with the uncertainty of both the raw input information and propositions deduced from it represents a substantial challenge without any consideration of coordinated attempts to deceive.

The extreme intricacy and subtlety to which the tools of deception can be applied in warfare are not often appreciated by computer scientists. Winston Churchill said, "In wartime, truth is so precious that it must always be accompanied by a bodyguard of lies."

Even trained intelligence analysts dealing with raw unfiltered data can be fooled and fooled badly. By comparison, an automated intelligence gathering system, especially one which may have been compromised, is especially vulnerable to deception. By providing such a system with a set of inputs designed to match its heuristics for making a decision, an enemy could cause large perceptual mistakes in automated systems. Such systems could be considered information amplifiers, which multiply the impact of our knowledge, but by the same token could amplify the disinformation provided by an enemy. Methods for detecting and dealing with spoofing must be considered in designing any such system, and ideally operators should be able to assume any task for which the system is suspected to malfunction.

Other AI applications systems, such as communications network management, are not as strongly affected by being compromised. They perform a narrowly defined task of monitoring and control. After an

attack, the network manager needs to survey the available subnetworks and communications channels to rapidly reestablish connectivity and assess damage. This task can be complex, involving considerations of atmospheric and ground propagation environments and use of knowledge about the original network and subnets. Information acquired early in the reconstitution task can be used to guide its continuation. Vulnerability to compromise is reduced in this particular domain, because any device that will allow you to think that it is a jammer is, in fact, effectively a jammer and any node that appears to be missing, is, effectively, missing.

These two example applications, the intelligence integration system and the communications network reconstitution and management task, illustrate two classes of potential AI applications--those subject to deceptive countermeasures and those that are inherently not "spoofable."

Applications deemed vulnerable to deception are thus uncertain candidates for knowledge based systems. For these domains, the level of automated decisionmaking should be limited. It is better to provide an "intelligent assistant" who is proficient at locating and presenting information from diverse sources [4,16,30] than an "expert" proficient at more analytical tasks.

The preceding examples attempt to illustrate some of the subtleties arising in military applications of AI that do not necessarily have direct analogs in civilian or commercial domains. In the classic *On War*, Karl von Clausewitz wrote that information is

> in fact, the foundation of all our plans and actions. Let us consider the nature of this foundation, its unreliability and uncertainty, and we shall soon feel what a dangerous edifice war is, how easily it may fall to pieces and bury us in its ruins.... A great part of the information in war is contradictory; a still greater part is false; and, by far, the greatest part is somewhat doubtful. It requires that an officer possess a certain power of discrimination, which only knowledge of men and things and good judgment can give.

V. AIR FORCE APPLICATIONS OF EXPERT SYSTEMS

A SAMPLING OF POTENTIAL APPLICATION AREAS OF MILITARY SIGNIFICANCE

The examples which follow are intended neither as a complete list nor as a recommendation of which problems should be deemed as the most significant. Rather they constitute a set of obvious targets of opportunity. While reading these examples, the reader should consider their application to his specific task area.

Expert Systems for Command Decision Aids

The most common products of knowledge engineering are consultation systems. Hence, one early application area attracting interest is consultation systems for military domains. Commanders typically must make crucial decisions under time pressure. Their decisions involve situations of great complexity with incomplete information. Thus there is ample need for advisory programs embodying compiled expert wisdom to provide suggestions and to fill in details once the overall direction is decided upon. Among the functions that such a system might perform are data analysis, situation assessment, and planning.

There are two types of system which could be constructed. The first provides to each commander advice based on the compiled expertise of the best domain experts which can be obtained. The second type leaves expert judgments to the commander himself. Instead it takes care of many details which any commander would understand, but whose automation frees the user to devote his attention to more important concerns. The techniques of knowledge engineering allow for the automation of many functions not feasible with other methods of software engineering. Thus this programming technology in itself constitutes a great part of the potential of this field.

Systems of this sort are already under development [4,7,16,25,29]. Further advances in technology will allow a much wider range of applications to be addressed in time. The chief barrier to the deployment of these systems is not, however, simply technological. There is also the lack of experience in the design of these systems for

the field. Not enough is known about what factors will be critical to acceptance and usability of these aids. As with other sorts of automation, the ability to compute certain information may not be sufficient.

Hardware Fault Diagnosis

With the increasing complexity of military systems has come a growing problem with reliability and repair. Repairing highly complex machinery and electronics requires very highly trained technicians, who are in short supply. The time required to repair important and expensive equipment is important to the fighting ability of all branches of the armed services. In particular, the repair of aviation subsystems, both mechanical and electronic, is an important application area. Hardware fault diagnosis is a highly skilled task, difficult to teach or codify in a manual. Hence, expert systems with the ability to perform hardware fault diagnosis would be of tremendous benefit. For certain cases this may be accomplished in the fashion of present day expert systems, by using shallow knowledge to connect symptoms with likely causes. However, the combinatorics of possible failures often make such an approach impossible. Additionally, such systems would be quite inflexible in the hardware it dealt with. A promising approach is to incorporate deeper knowledge of the connection of structure to function. A system with the ability to deduce function from structure could predict proper behavior from the hardware description, and given symptomatic behavior, suggest possible causes and tests to disambiguate them. There are research efforts directed toward producing such capabilities in a number of locations [10,19].

Hardware Maintenance Consultant

A similar but larger concern is the entire area of hardware maintenance. The conflict between increasingly more complex (and expensive) equipment and the limited number of skilled maintenance personnel will continue to worsen unless novel approaches to the problem are discovered. This is a highly leveraged application where even a small improvement on equipment outage could be significant. Thus a very tempting application of knowledge engineering technology is the creation

of a maintenance assistant. Such a system would give the maintenance technician access to varying sorts of advice, both technical and heuristic. Such a system would serve the function of technical manuals, containing all such information in a more accessible form. They could also perform advisory and assistance functions, performing diagnosis, suggesting courses of action, and possibly directly collecting data. Such a system could allow much less skillful personnel to function adequately in this role. Such systems could be sufficiently sophisticated to make all decisions in common cases, requiring its human partner only to do the physical manipulations and report observations. Such very intelligent systems are unlikely to be realizable in the near future, but more limited versions may still ease the task of maintenance personnel. Some steps in automating the maintenance process may be feasible immediately.

Intelligent Database Retrieval

Intelligent database retrieval is an area of great importance for Air Force and other military applications. Few organizations can claim to maintain as large and diverse a set of databases as the DOD. They range from manually maintained file systems to extensive automated systems with nearly real-time maintenance. A system which could answer high-level questions by accessing the many databases would represent a valuable asset to military planners and commanders.

Machines That Assist in Their Own Use

By incorporating on-board processors running knowledge based programs, equipment may be designed to interface with its users in extremely flexible ways. One very interesting possibility of this sort is the creation of intelligent avionics. The pilot (or aircrew) of a combat aircraft is faced with a large array of tasks. The priorities and difficulties of these tasks are changing dynamically throughout a mission. The division of the workload between man and machine ideally should be equally dynamic, with the man taking the most important tasks and the intelligent machine handling those which the men are too hard pressed to attend to.

Machines That Explain Their Own Function

The increasing complexity of the machines we are building also creates difficulties for operations. Each new system requires increasing amounts of training to prepare the operator. Accidents resulting from operator error are also more likely as the complexity of the operator's job increases. These difficulties could be eased by the use of knowledge based man-machine interfaces with expertise about the operation of the device. Such a feature could be added after the original equipment is built. However, smart hardware to assist the operator and guard against critical mistakes is best achieved by incorporating the concept of an intelligent user interface from the beginning of the design process. A fundamental barrier to this application is the current high cost of producing a knowledge based system. If the improvement of this process results in a significant lowering of this cost, such applications will become extremely viable.

This concept is equally important when applied to computers. Computer programs, operating systems, and computer models that could explain themselves would be much easier to use. They would also be less likely to be misused. Rule based systems show promise for bringing progress to this area.

Intelligent Simulations

Numerical simulations to study various phenomena, and to provide test-beds and training environments, are well known. Knowledge based programming opens up broad new areas for simulation which previously had been very difficult to capture. This technology makes it possible to model the logical connectedness which is common in the structure of human institutions, artifacts, and behavior. Such simulations could be used for training, aids to decisionmaking, or tools for policy analysis [21,31]. An example of an area where a simulator might prove useful is electronic warfare. Many EW environments are poorly understood due to the closely guarded nature of the systems, plus the disruptive effect on civilian operations of turning on high power military electromagnetic emission systems for exercises or testing. A C^3I simulator could operate to evaluate a plan over a range of EW assumptions regarding the

effectiveness of both sides' EW measures and countermeasures. This approach would also be useful in training since real environment cannot be recreated [1].

Knowledge Based Library

One of the central problems with human expertise is that it is volatile. An expert can at any moment become unavailable either permanently or temporarily for a variety of reasons. When an institution has a great deal of resources invested in the expertise of its personnel, attrition can constitute a substantial loss. This is especially true in the military where in addition to death, illness, and retirement, there are losses from rotation and promotion. The totality of expertise which is available will never be captured in computerized form. However, that which is capturable could constitute a very valuable resource if the expertise of various individuals can be compiled so that a long-term organizational memory is created. Such a facility would assist new staff to spin-up to a level of expertise more quickly. Implementing this idea must await further progress in knowledge acquisition techniques.

PROSPECTS

All of these examples (with the possible exception of the knowledge based library) are potentially realizable within the next decade. Command decision aids and intelligent simulations already exist in prototype form [4,7,21,30]. Hardware fault diagnosis is being pursued in several different locations [10,19]. A hardware maintenance consultant of a primitive sort could be produced with today's technology. It could display likely pages of electronic manuals, based on the external symptoms. It would essentially be a data base manager for the large number of relevant manuals. A much more useful maintenance consultant will eventually be possible, incorporating hardware diagnosis abilities and very facile user interfaces to conform its actions to match the abilities of its user. Machines that explain their own function are essentially possible now, limited primarily by cost. A knowledge based system is more expensive to produce than is more traditional software. Thus the incorporation of this technology

would be warranted only for very expensive devices. If it becomes cheaper to produce such systems, their use could become much more general.

RISKS AND OPPORTUNITIES

Although there can be little doubt that systems of this sort can be realized and will be of use, caution is needed in judging whether the anticipated benefits justify the costs of developing any particular application. The chief risk is in development time and cost. There are additional uncertainties in how widely this technology may be applied, which will be a factor in deciding its real importance.

The expert systems developed for military domains up to this time have been laboratory studies, not engineered for actual field use. Prototype command decision aids exist and demonstrations are available, but it has not yet been demonstrated that such systems can be implemented in an operational environment. Unaddressed problems include:

- Online access to required data from operational data bases.
- Coverage for a realistic range of military situations and details.
- Implementation in an efficient software and hardware environment.
- Interface design for use by real personnel.
- Full system tests under realistic conditions.

That such details can be worked out seems very likely. However, estimates are lacking for the level of difficulty, and hence the cost, of such development.

The knowledge engineering work to the present time has been done by highly skilled personnel, working under very supportive academic environments. The wide-scale use of this technology may be hampered by a shortage of sufficiently skilled personnel, by lack of experienced managers, and by the lack of tools to assist in the development process. The shortage of personnel occurs at several levels. There is still a great deal of basic research to be done in the application of knowledge

engineering techniques to the variety of problems which may eventually need to be considered. The highly skilled (Ph.D. level) individuals needed to conduct this research are certainly in short supply. Of equal importance to development are the programmers and software engineers which will be required to successfully develop products based on this technology. The programming incorporated into many of the existing expert systems is quite sophisticated, and the skills necessary to program at this level are rare. Large scale production of knowledge based systems must occur in more structured environments than those of the prototype efforts. A shortage of managers with experience both in knowledge engineering and in controlling a software engineering effort is likely.

Knowledge based systems produced up to this time have required large amounts of labor from individuals whose skills are in short supply. If this remains true, future applications will need to be limited to a few problems of paramount importance. For more general use to be feasible, progress must be made in several areas. First, adequate personnel must be trained. Second, standard approaches to the solution of knowledge engineering problems need to be abstracted so that staff with less creativity may produce useful products. Finally, tools to assist in the construction of expert systems need to be developed to enhance the productivity of the available personnel.

In summary, knowledge based systems have been demonstrated to be very powerful tools for the solution of at least some (and potentially many) important problems. It remains to be demonstrated how widely useful this technology will prove to be and how rapidly it can be practically applied to military domains. If the promise of this innovation is to be fulfilled, research must continue on several levels. Clearly distinguishable are:

- Basic research into fundamental issues of artificial intelligence (e.g., machine learning). These areas have high payoff but very long time to fruition.
- Research into long-range knowledge engineering issues. Examples include techniques for extracting knowledge from

experts and for constructing trusted knowledge based systems or trusted development environments for such systems.[1]

- Research into basic knowledge engineering technology, to allow the creation of systems of types not now possible. Less speculative than the above, these efforts have higher probability of success, and the results may be available for applications use within a few years.
- Development of prototypes of military domain expert systems similar in concept to existing successful research efforts. These studies involve knowledge engineering in the military domain.
- Development of tools to assist in the construction of knowledge based systems. This may include special hardware, knowledge based system programming languages, development systems, or programmer aids.
- Preliminary studies in the design of knowledge based systems for actual use, including human interface design and modifications based on field trials.

[1] A system or component thereof is said to be trusted if its hardware and software security safeguards have been constructed and tested to specific criteria to establish that the system (or component) will enforce a security policy with a certainty that is appropriate to the intended application.

VI. THE PROSPECTS FOR LARGE-SCALE KNOWLEDGE ENGINEERING EFFORTS

One of the key characteristics of present knowledge based systems is they must be applied to a strictly limited domain. It is this limitation which has made the recent successes of knowledge engineering possible. By keeping the domain limited, it has been possible to capture the necessary knowledge, devise a uniform representation for that knowledge, and to constrain the amount of searching required to utilize this knowledge.

However, just as conventional software engineering has concerned itself with the construction of enormous programs, we may speculate on the possibility of applying knowledge engineering technology to the construction of very large knowledge based systems. Currently, systems consisting of less than a few hundred rules are most common, systems with more than 1000 rules may be termed large. At this time, construction of a system utilizing 10,000-100,000 rules would be an extreme undertaking.

The size of a system can be measured in several ways--by the amount of knowledge it contains, by the complexity of the procedures it executes, or by the resources required for its construction. The critical question is the relationship between these measures. The most favorable answer would be one of linear relationship, where the cost of a system scales smoothly with the size of the domain addressed. On the other extreme, if the relationship is exponential (or worse, hyperbolic) the implication would be the practical impossibility of building systems larger than a limiting size.

While there is essentially no experience at present from which to extrapolate reasonable scaling laws, there are a number of arguments which suggest caution in applying knowledge engineering to poorly limited domains. As the domains become larger, the difficulty of capturing the needed knowledge with a single knowledge representation increases as well. Consequently, for very large domains, a number of different knowledge representation forms may need to be employed. Thus the software complexity may scale nonlinearly with the size of the

domain (except in the unusual case of a large domain containing very regular sorts of information). Additionally, although many knowledge representation techniques attempt to provide modularity of knowledge (new facts can be added independent of what is already known), this modularity is seldom perfectly achieved. Different pieces of knowledge may occasionally interfere during inferencing, which can create very difficult debugging problems and the need to fine-tune system behavior. Thus as the knowledge base grows in size, especially with different representation forms and more complex software, assuring consistency and completeness of the knowledge base may become a growing concern. Also, as domains grow so do search spaces, generally quite exponentially. Thus conceivably so could the meta-knowledge or software complexity used to constrain search. The complexity of knowledge based systems may grow faster than the domain covered, and the difficulty in constructing them could grow nonlinearly with the size of the system.

Another class of scaling problem is not purely technical but human or managerial. Knowledge based systems developed in the past have generally been the product of very small teams, holding to a minimum problems of management and communication. This is especially important in obtaining the required domain knowledge--a very difficult and time-consuming procedure, at least as it is now performed. In certain cases of past successful expert systems, the developer was also a domain expert. Even where this was not so, typically only one domain expert is the organizer of the knowledge being collected. Heuristics obtained from other experts are subject to the scrutiny of this "knowledge czar" for consistency with the existing scheme. In a very large system this would no longer be possible. Multiple teams of implementors and experts would be working on various subdomains. The difficulty of coordinating their efforts may obey a nonlinear scaling law.

Nonlinear scaling phenomena may be observed in connection with conventional software engineering. Although the phenomena have been successfully dealt with, they are significant factors in the costs of developing very large systems. Scaling difficulties may at least initially be much greater for large-scale knowledge engineering than with current software methodologies. Whether these effects will be permanent barriers or the early phase of a technological learning curve may ultimately be revealed only through experience.

A related concept is that of an open versus a closed domain. A large domain may still eventually be captured by knowledge based approaches if it is closed--that is to say, if its boundaries can be firmly defined so that only knowledge within those boundaries will be required for the system to adequately perform its task. On the other hand, even the smallest problem will be intractable if it is open. An open domain is one which on occasion may require an extraneous fact. This is the case with most so-called common sense reasoning problems. The core task may be quite simple, yet the need for information of an unpredictable form makes such areas very difficult.

The foregoing analysis should not be taken to imply that these problems are unsolvable. They in all likelihood will be overcome eventually. It is necessary, however, to be cautious in estimating the earliest time that a particular project will become feasible, and its probable cost. If there are going to be any massive disappointments from knowledge engineering, they will in all likelihood be a consequence of attempting to tackle too large a domain before the necessary techniques for doing so have been developed.

VII. KNOWLEDGE BASED APPROACHES TO COMMAND, CONTROL AND COMMUNICATION

Command, control, communication, and intelligence (C^3I) is a military application area needing innovative approaches. As new technology is introduced to the battlefield, the critical response times required for decisions are growing shorter at all levels of the command hierarchy. Communication devices and control hardware are constantly being improved to meet this demand, but the human elements of the system are unable to change their timing characteristics. Thus an ever increasing stress is being placed on human decisionmakers in the modern battlefield. Additionally, the C^3 facilities must be robust against potential counter-measures, and the C^3 network be designed for survivability and reconfigurability. The confluence of increasing information flow rates from advanced sensors, the growing need for speed, and the necessity of flexibility calls for increasing levels of automation of this facility. Computers must take on tasks which were originally done by humans as well as other tasks created by the modern environment. Since these tasks are predominantly nonnumeric, characterized by very nonregular information, and lack algorithmic procedures, knowledge based approaches seem a possibility.

Among the potential applications of knowledge engineering which may address command, control, and communication problems is the use of expert systems to assist decisionmakers. This application has been considered above. Such systems could advise on possible options, fill in plan outlines, and assist in administrative details. This application does not differ appreciably from other potential uses for expert systems. Such systems will in all likelihood prove very useful, and could be initially deployed within the coming decade. In this section we shall consider two more aggressive uses of the technology--knowledge based networking and sensor fusion.

The division of knowledge based applications into those vulnerable to "spoofing" versus those less vulnerable to deception was discussed previously. It should be emphasized that this distinction is important in considering knowledge based C^3 applications. A related perspective

is to distinguish top-down from bottom-up applications. Top-down applications are those used by a high level command staff to manage the mass of information available to them, whether it is gathered by conventional or computer assisted means. Bottom-up applications are those that interface with the "real world" in various ways and semi-autonomously perform actions, reduce data, and provide input to higher levels of the hierarchy.

Top-down approaches to knowledge based systems for C^3I provide command decision aids for command personnel which interface them to the C^3 network. Determining where to look for a particular piece of information is a nontrivial issue. Actually retrieving it can be a substantial task which complicates and slows the process of command decision. (An example of this type of interface is in Ref. 24.) An automated system could also decompose higher level queries into their components with respect to the existing databases and file systems, and to allow alerting over multiple databases. This consists of periodically looking for developments meeting some set of criteria. Once decisions have been reached, the same devices may allow abstract orders to be automatically filled out and decomposed into components which are then distributed to various destinations.

Bottom-up applications, such as sensor fusion and communications network management and reconstitution, interface with the actual physical environment and provide input to the databases. Sensor fusion involves correlating, merging, and interpreting the inputs from distributed sensors in the field. Network management systems would be electronic agents, charged with the responsibility of managing the C^3 resources in response to varying load, shifting priorities, and possible hostile interference.

SENSOR FUSION

A wide variety of sensors may be used in the identification and disambiguation of the numerous different weapons systems, platforms, vehicles, and installations in the field. Combinations of sensors may be employed to yield information not available from any single source. Examples of information sources include infrared sensors, radio receivers, cameras, motion sensors, communications intercepts, readings

from position location equipment, radar, satellite data, and reports from human agents. The fusion task depends on both knowledge of the environment to be observed and the priorities for presenting information to the human members of the team. The methods for combining data may vary with the situation, environmental conditions, and knowledge about operational priorities. The flood of data available may very quickly overload human analysts, even with present day computer assistance. A sensor is of no use if its data are overlooked or ignored. Intelligent systems could process the raw data, using knowledge about how the information is best utilized, and present refined conclusions to the human components of the system.

Fusion could take place on a variety of scales, ranging from a few sensors connected to a weapons system up to a theater network. The fusion application is sensitive to countermeasures based on deception and would profit from further theoretical work on decisionmaking in uncertain, and in fact, hostile circumstances. The penalty for errors by a sensor fusion system increases with the level of conclusion drawn. The developmental strategy for such systems should be to gain experience at small scales before attempting more aggressive implementations.

Automatic sensor fusion at any scale must be approached cautiously. That the system be accurate in its identifications is of course critical. The characteristics of enemy targets as well as enemy countermeasures such as jamming or spoofing may change after the system is built. Thus it is necessary that such a system be designed for flexibility. The human operators should have as much freedom as possible to alter system characteristics and to access data at varying states of refinement. Of particular concern is that the compromise of such software would allow the enemy to design countermeasures tailored for its specific fusion heuristics. Thus it will be critical that human operators remain "in the loop," able to monitor and override system components as desired. No knowledge based system has yet been demonstrated to deal successfully with deceit. Human agencies acting with deception may find an "artificial intelligence" particularly easy prey.

The degree to which sensor fusion is vulnerable to enemy countermeasures varies depending on the particular application. The coordinated control of groups of semiautonomous robot drones or ground vehicles has a relatively lower vulnerability than other applications previously discussed. An operator of such a fleet could be rapidly overloaded by the raw output of cameras, which would be vibrating, of varying focal lengths, and pointed in different directions, in addition to the outputs of other sensors. A sensor fusion and integration system could take the raw data and the history from each sensor, combine it with terrain mapping and other knowledge about the situation, and provide an integrated synthesized view for high level control. This "bird's eye" view could be put anywhere. It would be a synthesized view and thus not subject to being obscured by smoke, clouds, or other phenomena. Such a system would also be able to identify areas based on old information, although it would not be given the same credibility as current observations.

The same aids developed for sensor fusion and command decision can be used for training by providing an artificial environment. Training applications are attractive given the constraints on use of electronic warfare systems and exercises (i.e., trying to avoid civilian radar blowouts or broadcast interference) and in simulating the use of weapon systems which may be too costly to spend in exercises.

DISTRIBUTED KNOWLEDGE BASED SYSTEMS

An important aspect of the command and control system is the communications network which allows orders and information to be distributed among sites. This network is in need of management, both in distributing information and in reconstituting itself when damaged. The requirement that C^3 systems be reconfigurable implies strong flexibility constraints on the communications network.

The communications network management task, described earlier, is a specialized version of the sensor fusion application. Here the sensors are those monitoring the availability of communication channels during the course of an engagement. The inherent lack of vulnerability of this task to deceptive measures makes it an attractive candidate application.

A communications network may be used for more than an advanced telephone system. By connecting various computerized stations it becomes possible to solve problems using knowledge and information from a group of sites. This can be referred to as distributed problem solving. A simple form is to interface command decision aids so that they may exchange information and problems. Thus the architectures of the local systems are to some extent independent. They share a model for requesting information or assigning tasks to one another. The means for controlling such a distributed approach is an active research topic, and there is already some indication that such systems should be feasible. A command decision aid can also become the local interface to the command, control, and communication system.

A more aggressive design would be a fully distributed system that took advantage of all of the concurrency available on the network. Such a system might use multiple processors, multiple sensors, and multiple knowledge sources. The processing burden could be distributed dynamically in response to shifting loads, facility losses, and communication link outages. In this design the local nodes of the network would need to be much more homogeneous, so that the entire system may be regarded as a single distributed knowledge based system. This is a level of difficulty greater than the more modular approach.

An extreme version of the network would take humans totally out of the loop. Advanced design computers, exploiting sensors and using weapons in a closed loop, would constitute a single enormous weapons system. While this idea would solve response time problems, the technical difficulty of ensuring the correct action of such a system under fire would be enormous. Although such a proposal may not be technically impossible, this model of the future of the technology should not be encouraged. The necessity of dealing with the unexpected constitutes an open domain problem. Consequently, there is an enormous premium in maintaining men in the loop with full potential access to any and all information and control features. Rather than replacing human personnel, the object of the technology should be to augment their capabilities. It seems virtually certain that for the foreseeable future there will be many cognitive tasks at which men will far

outperform any machine. Thus future systems should be designed with the goal of achieving an effective partnership between men and machines. The ideal system would adjust the division of labor between man and machine in response to external conditions. Here one is tempted to speak not only of a man-machine partnership, but more graphically of a man-machine symbiosis.

SUMMARY

It is clear that C^3I is a very interesting applications area for knowledge based technologies. Among the forces driving this interest are the increasing information overload suffered by personnel in the modern battlefield, and the requirements for C^3 system survivability, reconfigurability, and interoperability. The introduction of knowledge based systems may augment human performance and improve flexibility of the total system. This flexibility may be exploited to help the system adapt to changing requirements as well as deal with the unexpected.

Command decision aids to assist in C^3 related functions could be introduced in the near future. The technology is sufficiently advanced to allow the automation of at least some local functions. Additional tasks could be automated gradually over the next several decades. The key applications of large-scale sensor fusion and knowledge based networking are clearly of a higher order of difficulty than the encapsulated expert systems. Any initial work in these areas must emphasize the identification of subproblems which have limited complexity and strong resistance to spoofing. However, knowledge based technologies cannot be expected to make an immediate contribution to any large-scale applications. An evolutionary strategy for development involves the least risk. A commitment to a universal system based on knowledge engineering should not be undertaken until much more experience has been gained in the strengths and limitations of this approach.

VIII. ADVANCED SENSORS AND INTERFACES

Associated with the field of artificial intelligence are a number of disciplines which do not fit under the rubric of knowledge engineering. They include such topics as machine vision, natural language understanding, and robotic applications. They are connected with knowledge engineering not only for historical reasons, but also because they are potentially important application areas for this technology. In each of these areas, fundamental processing issues must be addressed which are not knowledge based. However, knowledge based approaches are of interest in relation to more advanced problems. Consequently, some of the greatest prospects for knowledge engineering may materialize after progress has been made in the associated disciplines, and near-term progress depends on domain-specific research. Detailed assessments of these varied disciplines are beyond the scope of this paper, but no survey of potential applications of knowledge engineering would be complete without mentioning these advanced possibilities.

MACHINE VISION

Image understanding by computer has enormous potential significance, particularly to robotics. Other potential applications include satellite and aerial image interpretation, surveillance, and vehicle and weapons guidance.

Current systems can identify objects in uncomplicated settings, with simple backgrounds [29]. Complex scenes with object occlusion and other difficulties are very demanding for systems running on large computers. Smaller systems suitable for robotic application are primitive, requiring a very structured environment to perform acceptably. Progress in this field is certain, but the difficulty of the obstacles is hard to estimate. Fundamental research is needed in the principles of vision. Progress in this area is decoupled from the general problems of artificial intelligence and substantial progress is possible with or without progress in knowledge engineering. High-level

machine image understanding will require the use of "common-sense" knowledge about the physical world. However, progress in the lower levels of image processing is possible using much different approaches. There are substantial problems remaining to be solved in the extraction of features from the raw image. The early stages of image processing are very amenable to highly parallel computations, consequently special-purpose hardware for image understanding is a very likely possibility [5,8].

NATURAL LANGUAGE UNDERSTANDING

"Natural language" processing includes interpretation of written text, speech understanding, and language translation, with the latter two significantly less developed than the first. One of the main barriers to the use of computers by untrained personnel is the highly stylized interface which the machines present. As machines become more capable of understanding natural language, it will become much easier to design machines that can be handled more easily by people. Other applications such as machine translation of natural languages or computer analysis of natural language texts would also be possible.

Unfortunately, it appears that the complete solution of the natural language problem is extremely difficult. General understanding of natural language requires large amounts of world knowledge and common sense reasoning [18,37]. Thus there is little prospect of general natural language understanding by machines in the foreseeable future. However, restricted subsets can be more tractable. Thus it may be possible to design an English-like language for communicating with machines. Such a language would appear as a subset of English to humans while avoiding confusing features which would inhibit machine understanding.

In the military context, language is frequently constrained and its constructs very stylized. The vocabulary is often limited, and grammatical forms quite simple. Consequently, there may be many potential military applications for current natural language understanding.

Expert systems often incorporate a limited natural language capability. Their usefulness would be enhanced by more flexible language and explanation facilities. Progress in these areas is being made at various research laboratories.

ROBOTICS

The prospects for robotics in military applications are exciting. This technology, once it has matured, could allow the creation of autonomous and semi-autonomous weapons systems with extreme versatility and intelligence. It is not too early to begin studying the possible uses to which these capabilities may be put.

In addition to battlefield applications, robots may play a major role in space. As the Air Force mission in space expands, an important application will be the maintenance of space systems using robots or teleoperated devices. This approach is attractive for a number of reasons. Servicing for space systems will be costly and difficult for satellites beyond the 600 mile orbital range of the shuttle. The use of a mobile robot/teleoperator (possibly maintained in an orbiting space operations center) would be a viable alternative to maintenance by crews. A number of technical issues, many of which fall in the area of artificial intelligence, must be addressed before such operations become feasible. In particular, it is necessary to endow autonomous systems with the ability to look ahead and evalu te their actions. This is especially important in space where long delays in the control loop require that the system be able to operate with only high-level guidance.

In the near term, progress in robotics will depend on developments in sensors (e.g., machine vision) and in the construction and effective use of effectors. Between these two, knowledge based technology has a contribution to make in situation assessment and planning [18,24].

IX. KNOWLEDGE BASED SUPERCOMPUTERS: FIFTH GENERATION MACHINES

THE NEED FOR SPECIAL MACHINES FOR KNOWLEDGE BASED APPLICATIONS

Knowledge engineering as we have defined it is essentially a novel technology for constructing computer software. Much of the research has been done on large computers which are designed to run quite general computer software. From this one might expect that future general-purpose computers would suffice for the needs of knowledge engineering and artificial intelligence research. However, there are several indications that this is not the case. A recent trend in facilities for AI research is toward fairly expensive single-user workstations which are constructed to facilitate symbolic processing. Many of these are "LISP-engines," hardware constructed specially to run programs written in LISP. The Japanese have embarked on a massive project to construct a new generation of computers specially designed for knowledge based applications. Many members of the artificial intelligence community are also involved in the rapidly expanding investigation of novel computer architectures.

Special machines may be critical to the development of knowledge engineering. Most knowledge based programs tend to consume voracious quantities of processor time and computer memory. At this time, the task of producing a nontrivial knowledge based system for real-time applications is at least extremely difficult and is genuinely impossible for most jobs. Additionally, the process of constructing a knowledge based system differs from that used for more conventional software, and can benefit from hardware designed to accommodate this use. Finally, and perhaps most significantly, knowledge based codes show large amounts of potential parallelism. They thus may be especially suitable for implementation on some highly parallel computer architecture.

The machine-intensive quality of most knowledge based programs is due to several factors. Most primitively, prototypes in any technology tend to be constructed with emphasis on finding some solution, not on finding an efficient one. Thus some speed-up may be expected with the maturation of the field. There are, however, more fundamental difficulties.

First of all, many knowledge based techniques search to select information from a range of possibilities. Search methods can be very expensive computationally. The size of the space to be searched typically increases exponentially with the size of the problem which is to be solved. This is the reason why search must be constrained and why programs exploiting search methods will be computationally intensive.

Additionally, knowledge based programs typically exhibit less locality of reference than conventional computer software. Many modern computers use a virtual addressed memory to facilitate programs using very large amounts of storage. The efficiency of virtual memory (as implemented by paging of the address space) is very dependent on the locality of reference characteristic of most computer programs. As knowledge based systems often have low locality of reference (since any fact may be needed next) they often run slower due to the demands of the paging hardware of the virtual memory.

More generally, knowledge based systems must deal with highly irregular and dynamic information. Consequently, the languages for knowledge based programming, the environments in which the programming is done, and the resulting product must be flexible and powerful. This flexibility is provided by mechanisms that imply a loss of efficiency. The power and flexibility of LISP has made it a popular programming language in which to implement knowledge based systems. Because of its extreme power, LISP implementations are nearly always interpreters, which makes the resultant programs much slower. The dynamic nature of knowledge based programs requires large amounts of temporary storage, and consequently an active storage management process, both of which absorb resources. Simply the need to store information in flexible forms makes accessing that information more time-consuming than in conventional processes. Thus it seems clear that knowledge based programs will always make greater demands on computing facilities than more conventional software.

The demand for great flexibility may by itself motivate the use of special architectures for knowledge based programming. The move to high-powered work stations has been motivated not only by the greater speed of LISP machines but also by the very powerful user interfaces provided.

The difficulties involved with creating innovative knowledge based programs motivate the use of specially designed man-machine interfaces to ease the task.

From all of the above one might expect that knowledge based programs inevitably will be slow and expensive. There are some reasons to hope otherwise. First, the continuing evolution of computer technology may eventually allow programs which are currently too slow to run at quite acceptable speeds. Additionally, a large part of the inefficiency of knowledge based procedures may be due to a mismatch between the requirements of the program and the characteristics of the machines on which they are executed. Most computers are designed around the Single Instruction - Single Datapath (SISD) architecture, sometimes referred to as the von Neumann architecture after John von Neumann. This phrase implies a single processor sequentially executing a program which can access only one data value at a time. The linear memory model used by most classical machines suits well processes operating on arrays of data. Knowledge based processes, on the other hand, typically work with data that are structured associatively. A different memory architecture might conceivably support AI programming much more efficiently. The sequential execution of the classical SISD machine fits well with the step by step nature of most algorithms. But AI programs are seldom algorithmic, and may thus have large amounts of potential parallelism. In particular, search methods allow for parallel execution. There is therefore reason to believe that highly parallel architecture might support very rapid execution of some knowledge based programs. It may be possible to build machines which are much better suited to the uses of knowledge based systems. What these machines should look like and when they can be expected are a matter of dispute.

VLSI OPPORTUNITIES FOR HIGHLY PARALLEL ARCHITECTURES

The computer revolution of the last twenty years has been largely driven by the enormous progress during that time in computer hardware. Today's hand-held calculators selling for tens of dollars have computing power equal to the original digital computers which filled buildings and cost millions. During the 1970s the time required to reduce circuitry's size and cost by a factor of two was typically less than three years.

The next step in miniaturization is Very Large Scale Integrated (VLSI) circuits, semiconductor chips containing many discrete elements each less than a micron in size. Their introduction holds the promise of yet greater cost reductions and performance increases. However, the design of products using VLSI technology has proved somewhat problematic. We have already reached the point of being able to construct a computer on a chip. The prospect of being able to place one hundred times as many components on a chip stresses our ability to do so usefully. The design rules for VLSI devices may prove to be much different from those of the past. We are entering an era where processors are very cheap or essentially free. The problem becomes how to bring them to bear on work to be done.

The classical model of the computer, involving a single processor serially executing a program, was devised at a time when computer processors were very expensive. In the coming era, the cheapness of processors will promote the use of non-von Neumann architectures using multiple processing elements. Architectures in which multiple processors work in parallel would allow large increases in effective computing power if all the processors can simultaneously do useful work. It would seem that this should be possible. The human brain may be regarded as an information processing device. Its switching elements are a thousand times slower than those in today's computers, yet it is vastly more capable because it uses massive levels of parallelism. Massive parallelism may be very wasteful of processing elements on the average, but if processors are cheap this may be a viable approach to high speeds. Yet the design of a useful highly parallel computer has proved to be difficult.

Only a few of the algorithms in common use have very much inherent parallelism. One notable example is the manipulation of large arrays in numerical applications such as in signal processing and numerical simulation. The parallelism available in these processes has led to the most common sort of parallel architecture, the pipelined array processor. The parallelism there is all numeric.

There are a number of forces increasing interest in parallel architectures. One is the desire to use concurrency to enhance performance. Another is the opportunity provided by VLSI technology.

One possibility for exploiting VLSI is to place multiple processors on a single chip in such a way that they all work in concert. Classical architectures are ill suited for a number of advanced very-high-level programming languages, many of them developed for knowledge based applications. Often they serve as the motivation for a particular novel architectural design.

OBSTACLES TO THE USE OF VLSI

The tremendous advances in computer technology over the last twenty years have had such a large impact on society that the ability of the semiconductor industry to make continuing revolutionary advances has become a part of conventional wisdom. Graphs of price and performance vs. time appearing in various publications are commonly extrapolated to predict future devices with even greater performance. There seems to be no physical barrier to at least two more cycles of increase in device integration level. But an increase in integrated circuit (IC) device complexity does not necessarily lead to clear product advantages. At the present time the semiconductor industry is not process-technology limited for nonmemory products [27]--that is to say, more complex chips could be made, if we could design devices to advantageously exploit that complexity. The problem to be solved to use VLSI is what should be put on a Very Large Scale Integrated chip.

There is a precedent to this situation. Between 1965 and 1968 there were few computer products whose complexity came close to the limit of the time. One reason was that as the complex digital systems of the day were partitioned into blocks for IC manufacture, each block tended to become unique. Consequently, there was a large number of different parts with small demand for each. A second problem was that as the number of components on a chip rose, so did the number of leads required. Thus the usefulness of the semiconductor technology was strongly limited by packaging technology, which would allow only a small number of leads to a chip [27].

These problems were not solved in 1968; the work toward ameliorating them continues to this day. The problem of using new levels of integration was solved not by successfully applying the new technology to older products but by defining new products which fit the

possibilities inherent in the newly available complexity. In that case, the new products were hand-held calculators and semiconductor memory devices. Thus the interconnection and device definition problems have not been solved, merely circumvented. In the case of memory this is extensible to each new level of integration scale by continuing to make higher and higher density chips. For processing elements each new level of complexity requires a new product to exploit it. This role is being served by microprocessors of growing scale.

This history sheds light on likely future applications of VLSI. Memories of greater capacity can (and are) being designed. However, in the case of processing elements, it is unclear whether processors of the traditional sort can be designed to usefully exploit million component chips. To use this complexity, new products need to be developed. The special-purpose processors being developed under the VHSIC program are one possibility. Highly parallel processor architectures may provide yet another option.

THE JAPANESE FIFTH GENERATION COMPUTER PROJECT

Of the various investigations into today's novel computer architectures, certainly the most famous and the most clearly identified with artificial intelligence is the Japanese fifth generation project. In 1981 the Japanese Ministry of International Trade and Industry (MITI) together with Japanese industry launched a multi-year project to develop a so-called "Fifth Generation Computer" by the 1990s [28]. This project has been much discussed since then in the popular media [15]. Regardless of the likelihood that this particular project will succeed in its stated goals, the plan for the project captures a general vision of the future of computing technology, and it is nearly certain that progress toward these goals will continue to be made, in Japan and elsewhere.

The goal of the fifth generation project is most generally to build a supercomputer for knowledge based processing. The intention is to develop families of such machines spanning all levels from small processors to large scale supercomputers. The software for these machines span the list of significant open problems in computer science, with emphasis on artificial intelligence research.

The Japanese plan calls for completion of prototypes of the fifth generation computer system by the target year of 1990. Given the large number of outstanding unsolved problems on the list, the likelihood of complete success would appear vanishingly small. However, the advantages which might accrue from a partial success or from the spin-offs of a useful failure make the project of definite interest and consequence for the likely path of future technological developments.

The importance of the Japanese initiative lies not in the details of the plan, which may not be the best approach to developing knowledge based computers, but in the timeliness and boldness of their effort. Almost certainly they have correctly identified knowledge based software as an important component of the next cycle of major innovation in computing. Their ability to create a national project at this time to exploit this insight can only be admired.

The significance of a new generation of knowledge based machines lies both in reducing development time and in increasing the range of potential applications. Machines specifically designed for knowledge based applications could provide a given level of performance with a lower cost and smaller size than would be possible with conventional equipment. This will open up new potential uses. The same factors will allow more powerful equipment to be available to development personnel for the same price. This may be a significant aid in encouraging rapid development of knowledge based systems.

SPECIAL-PURPOSE HARDWARE FOR KNOWLEDGE BASED APPLICATIONS

From the preceding discussion, it follows that development of knowledge engineering technology will entail the creation of special-purpose hardware. LISP machines for artificial intelligence research have already emerged. Continuing evolution of general-purpose AI machines can be expected. Such machines may be fielded to run applications and thus become more than research tools. The Japanese vision of "fifth generation" knowledge based machines having wide application in diverse domains seems certain to be realized eventually, although their goal of the early 1990s seems optimistic. These machines are likely be available in great numbers by the late 1990s at the earliest, although it is difficult to predict computer technology.

There is another alternative to the general-purpose machine. Special knowledge based functions of great utility could be implemented on dedicated hardware. A current example is special image processing hardware. Highly parallel feature recognition hardware could be connected as a peripheral device on a conventional machine to yield a hardware basis for a machine vision system. A similar approach could be used in other applications of high utility. Clear candidates include continuous speech understanding, general natural language understanding, data retrieval, and inference. Continuous speech understanding and natural language are of such importance that if an adequate algorithm could be devised for at least some level of preprocessing, its implementation in VLSI could be easily justified. Data-base machines already exist for business applications. Many approaches to knowledge based hardware propose interfacing with such a system. Hardware providing structured memory for a particular application could be connected to a conventional machine to provide significant speedup. Finally, highly parallel inferencing hardware could be connected to a mainframe much as an array processor is now. A program could be written on an existing processor but avail itself of the special-purpose inferencing hardware to speed up computationally intensive processes. Such special-purpose, knowledge based peripherals provide an evolutionary strategy for the integration of knowledge based applications with more conventional computational environments. They thus may have potential worth alongside general-purpose knowledge based machines.

Basic research in knowledge engineering must concentrate on finding some way to solve a problem, not necessarily an efficient or immediately practical way. However, in considering the applications of this technology to the real world, questions involving practical matters such as processing speed become extremely important. The application of knowledge based techniques to avionics, weapon systems, or robotics will require execution in real time. Applications such as command decision aids, while not quite real time, still operate under very real response constraints. Thus the successful use of this technology will depend to a nontrivial extent on the hardware available.

Consequently, a long-term development strategy must include hardware to support practical application. In order for software research to evolve into development, appropriate hardware to support eventual field use needs to be developed in parallel. In the mean time, research into high payoff areas, such as hardware to support machine vision, should be encouraged. As the field evolves, a balance between prudent seeding of fundamental research and opportunistic applications development should be a stable goal.

X. CONCLUSIONS

The potential for applying knowledge engineering to problems of military significance is quite high, and certain possible applications may be startling. However, to assess these prospects, and to make reasonable estimates of the difficulties involved, it is necessary to recognize that as a technology area, knowledge engineering is a particularly demanding subset of software engineering. It is heir to all the difficulties normally associated with software, plus several which are unique. Examples of the latter include not only software and hardware but also managerial issues. Questions of how to specify knowledge based systems requirements, and how to manage construction of them must eventually be addressed. Great care must be exercised against expecting too much too soon, or in overlooking the possibility of unforeseen difficulties or bottlenecks.

PROSPECTS VERSUS RISKS FOR APPLIED ARTIFICIAL INTELLIGENCE

Knowledge engineering is an important new tool for the application of computers. By its use, problems which heretofore have been unsolvable may be addressed. The benefits from this technology may be expected to grow as our facility in its application improves. Among the benefits are:

- Providing a means for collecting, storing, and distributing the accumulated knowledge of domain experts.
- Providing a technique for creating computer software whose behavior is too complex for algorithmic treatment.
- Allowing greater access to computerized resources by untrained personnel.
- Allowing the creation and management of systems whose complexity might otherwise be prohibitive.
- Allowing the construction of mission equipment with much greater capabilities than are presently possible.

Because this is a novel technology, allowance must be made for unforeseen difficulties in its application to new domains. It seems virtually certain that the technology will eventually have a large impact across a wide front. What is uncertain is how much time and how many resources will be required before its promise can be realized. Among the risk factors which should be considered are:

- This is a relatively untested technology, and there may be unexpected problems in its application to any given domain.
- This is an immature technology. Successful application to a wide variety of potential domains requires further fundamental research, where progress is difficult to schedule.
- There are unresolved scaling issues, including how costs will scale with the size of problem. These issues are both technological (impact of domain size on software complexity, required processing power, and memory requirements) and managerial (whether the tasks of knowledge engineering can be standardized to allow large manpower efforts).
- Debugging and correctness verification may be more difficult than in traditional software engineering, or at least new approaches will need to be developed.
- Much growth in support resources is required before this technology can be widely applied. This includes development software and machines. The critical shortage will likely be skilled personnel including programmers, system designers, and managers.

On the whole, this technology area is very promising, and is deserving of continued support. There is a need to temper enthusiasm for the promise of this area with enlightened understanding of the problems which remain to be addressed. It is likely that there will be successful applications of knowledge engineering to specific problems in the very near term. Accessible problems from that point may be expected to grow gradually over a period of decades.

RECOMMENDATIONS

The application of research in knowledge engineering to military problems is an important endeavor requiring continuing support for research in artificial intelligence as well as support for the development of practical systems based on this technology. It is important to recognize that there are substantial engineering problems to be solved in bringing this technology to fruition. In order that fundamental research and practical development be encouraged to proceed in parallel, it is desirable to select key applications and encourage actual system development in these areas. Limited-domain computerized assistants and knowledge based simulations are classes of application for which there are demonstrations of principle.

As with any novel technology, there is the potential for initial disappointments. To avoid costly failures it is desirable that an evolutionary strategy be employed. Until the technology is more mature and better tested, it is best that applications be carefully chosen. The gradual refinement of techniques can be encouraged by sponsoring projects which enhance capabilities but which are not critical to the success of some greater enterprise. If there are any massive failures they will in all likelihood be the result of attacking too large a problem with too few resources. It is best that all initial problems be well defined and bounded. It is best if knowledge based systems be designed with the purpose of augmenting human capabilities, not of replacing personnel. The greatest benefit of this technology may be the creation of computer software which is sufficiently flexible that it can provide men with enhanced capabilities while preserving all of their options. This allows for total man-machine systems which have a maximum of robustness and flexibility, qualities which are at a premium in a hostile environment.

REFERENCES

1. *Air Force Magazine*, "What's Happening in Electronics at ESD," 1982, p. 74ff.

2. Backus, J., "Function-Level Computing," *IEEE Spectrum*, August 1982, pp. 22-27.

3. Barr, Avron, and Edward A. Feigenbaum, *Handbook of Artificial Intelligence,* 3 vols., William Kaufman, Los Altos, CA, 1982.

4. Benoit, J., A. Lemmon, and J. Selander, *KBS: An Expert Planning System for Crisis Response,* MITRE Corporation, RADC-TR-82-217, 1982.

5. Brady, M., "Parallelism in Vision," *Artificial Intelligence Journal*, 21(3), 1983, pp. 271-283.

6. Brown, Anthony C., *Bodyguard of Lies,* Harper & Row, New York, 1975.

7. Callero, M., D. Gorlin, F. Hayes-Roth, and L. Jamison, *Toward an Expert Aid for Tactical Air Targeting,* The Rand Corporation, N-1645-AF, 1981.

8. *Computer,* Special Issue on Computer Architectures for Image Processing, 16(1), IEEE Press, January 1983.

9. Cornish, M., et al., *The TI Data Flow Architectures: The Power of Concurrency for Avionics,* 3rd Digital Avionics Systems Conference, IEEE, 1979.

10. Davis, Randall, *Expert Systems: Where are we? And Where Do We Go From Here?* Massachusetts Institute of Technology, Artificial Intelligence Laboratory, A.I. Memo No. 665, June 1982.

11. Dennis, B., "Data Flow Supercomputers" *Computer*, IIEE Press, November 1980, pp. 48-56.

12. Douglas, J., "New Computer Architectures Tackle Bottleneck," *High Technology*, June 1983.

13. Duda, Richard O., and Edward H. Shortliffe, "Expert Systems Research," *Science*, Vol. 220, No. 4594, April 15, 1983.

14. Feigenbaum, E., *Knowledge Engineering: The Applied Side of Artificial Intelligence,* Computer Science Department, Stanford University, memo HP-80-21, 1980.

15. Feigenbaum, E., and P. McCorduck, *The Fifth Generation*, Addison-Wesley, 1983.

16. *Proceedings of the First Army Conference on Knowledge Based Systems for C^3I*, Ft. Leavenworth, Kansas, November 1981.

17. Gevartner, William B., "Expert Systems: Limited but Powerful," *IEEE Spectrum*, August 1983, pp. 39-45.

18. Gevartner, William B., *An Overview of Artificial Intelligence and Robotics*, Vol. 1, *Artificial Intelligence*, NASA Technical Memorandum 85836, June 1983; Vol. 2, *Robotics*, National Bureau of Standards, NBSIR 82-2479, March 1982; Vol. 3, *Expert Systems*, NBSIR 82-2505, May 1982; Vol. 4, *Computer Vision*, NBSIR 82-2582, September 1982; Vol. 5, *Natural Language Processing*, NASA Technical Memorandum 85635, April 1982.

19. Hartley, Roger T., "CRIB: Computer Fault-finding Through Knowledge Engineering," *Computer*, Vol. 17, No. 3, March 1984, pp. 76-83..

20. Hayes-Roth, Fredrick, Donald Waterman, and Douglas Lenat, *Building Expert Systems*, Addison-Wesley, 1983..

21. Klahr, P. and W. S. Faught, "Knowledge Based Simulation," *Proceedings of the 1st National Conference of the American Association for Artificial Intelligence*, 1980, pp. 181-183.

22. Martin, A., "The Torus: An Exercise in Constructing A Processing Surface," Conference on VLSI, California Institute of Technology, January 1981.

23. McCorduck, Pamala, *Machines Who Think*, Freeman, San Francisco, 1979.

24. Miller, Rene H., Marvin Minsky, and David Smith, *Space Applications of Automation, Robotics, and Machine Intelligence Systems*, Aramis Progress Report, MIT Laboratories, August 1982.

25. *Knowledge Based Systems as Command Decision Aids*, MITRE, ML-81W00003, 1981.

26. Mizell, David W., *Potential Supercomputer Architectures*, Computers in Defense Conference, sponsored by The Education Foundation of the Data Processing Management Association and Local Chapters, June 9-10, 1983, Los Angeles, June 13-14, 1983, San Francisco.

27. Moore, G., "Are we Really Ready for VLSI?" Caltech Conference on VLSI, January 1979.

28. Moto-oka, T. (ed.), *Fifth Generation Computer Systems*, North-Holland, 1982.

29. Nevatia, Ramakant, *Machine Perception*, Prentice-Hall, 1982.

30. Slagle, J., and M. Gaynor, "Expert System Consultation Control Strategy," *Proceedings of the National Conference on Artificial Intelligence*, 1983.

31. Steeb, R., and J. Gillogly, *Design for an Advanced Red Agent for the Rand Strategy Assessment Center*, The Rand Corporation, R-2977-DNA, May 1983.

32. Treleaven, P., D. Brownbridge, and R. Hopkins, *Data-Driven and Demand-Driven Computer Architecture*, ACM Computing Surveys, 14(1), March 1982.

33. Treleaven, P., "VLSI Processor Architectures," *Computer*, June 1982, pp. 33-45.

34. Waldrop, M. Mitchell, "Artificial Intelligence(I): Into the World," *Science*, Vol. 223, No. 4638, February 24, 1984.

35. Waltz, D., et al., "Artificial Intelligence: Assessment and Recommendations," *AI Magazine*, 4(3), 1983, pp. 55-67.

36. Waltz, David L., "Artificial Intelligence," *Scientific American*, Vol. 247, No. 4, October 1982, pp. 118-133.

37. Winograd, T., *Language as a Cognitive Process*, Addison-Wesley, 1982.

38. Winston, P. H., *Artificial Intelligence*, Addison-Wesley, 1977.

39. Woods, W., "What's Important About Knowledge Representation?" *Computer*, 16(10), October 1983, pp. 22-27.